

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

«Работа с файлами в языке Python»

ОТЧЕТ
по лабораторной работе №18
дисциплины
«Основы программной инженерии»

Выполнил:

Сотников Андрей Александрович

2 курс, группа ПИЖ-б-о-21-1,

011.03.04 «Программная инженерия»,

направленность (профиль) «Разработка

и сопровождение программного

обеспечения», очная форма обучения

(подпись)

Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

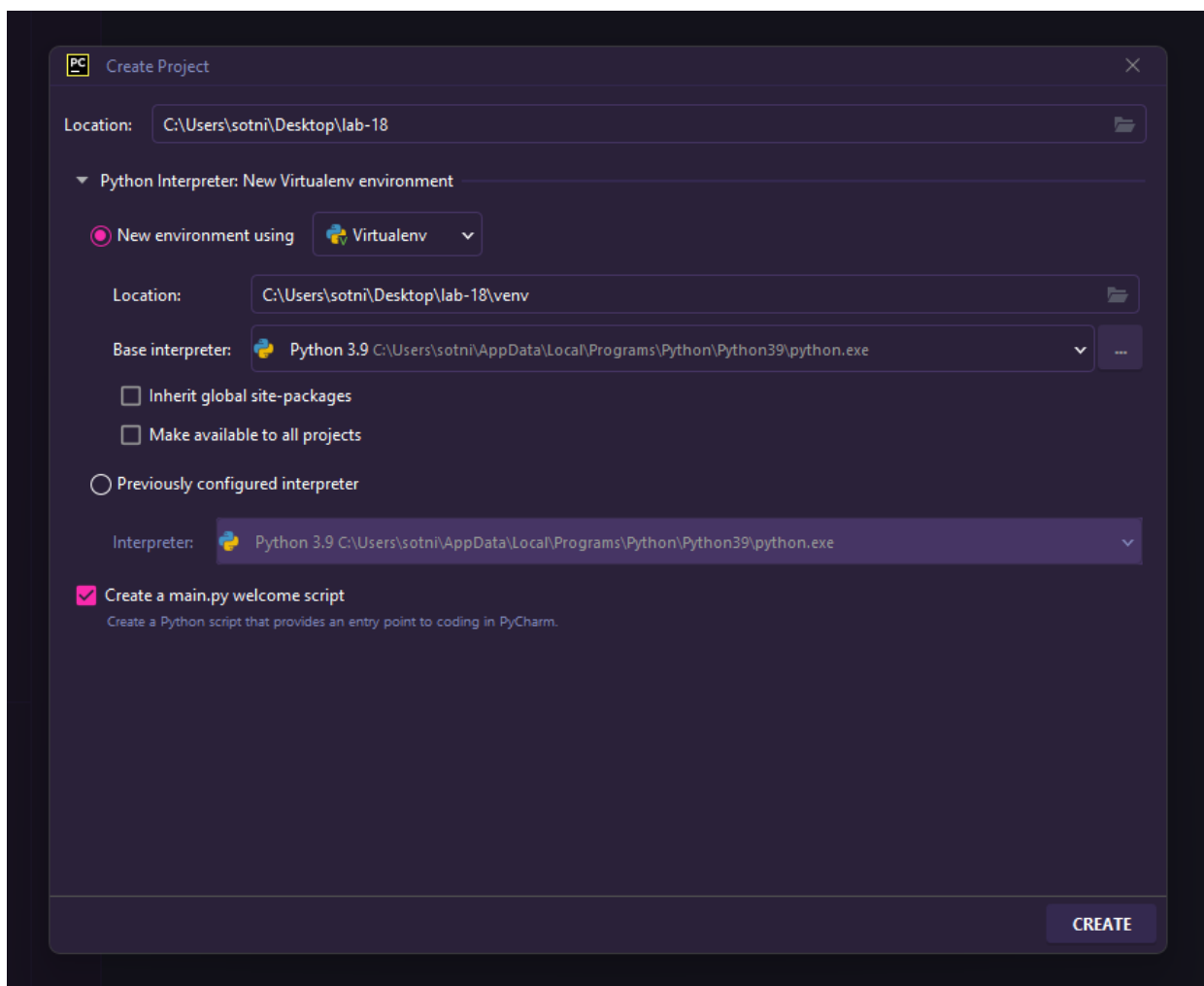


Рисунок 1 – Создание проекта и виртуального окружения

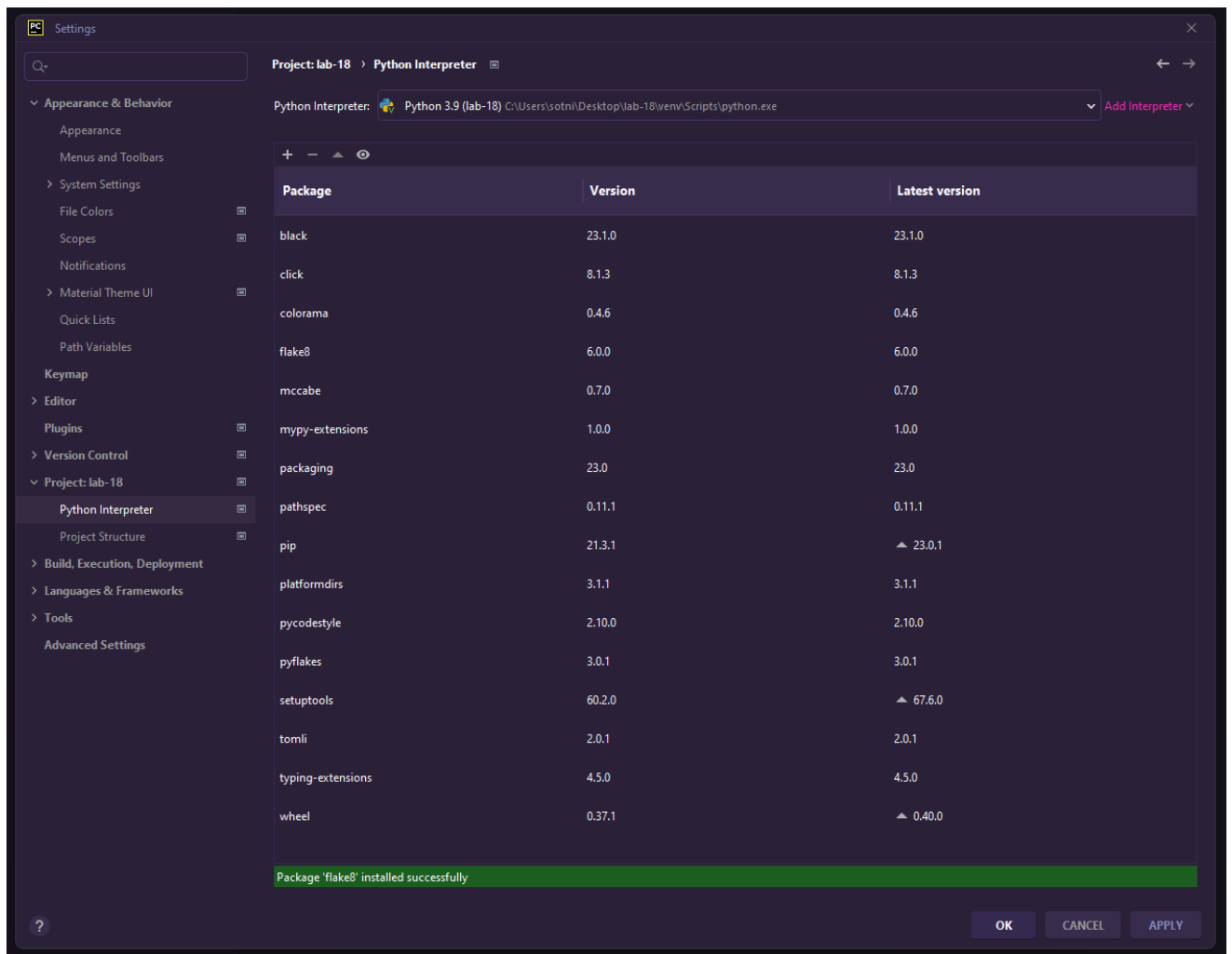
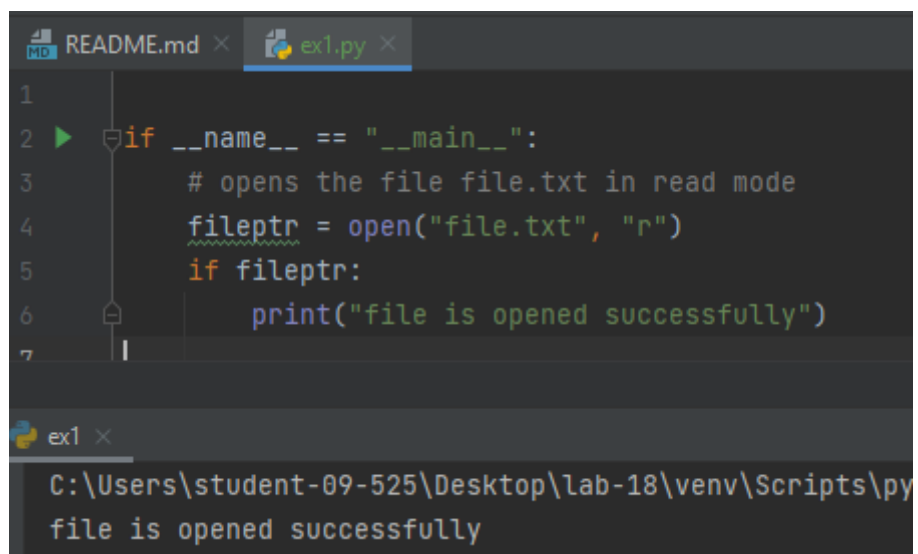


Рисунок 2 – Установка пакетов black и flake8

Проработка примеров:

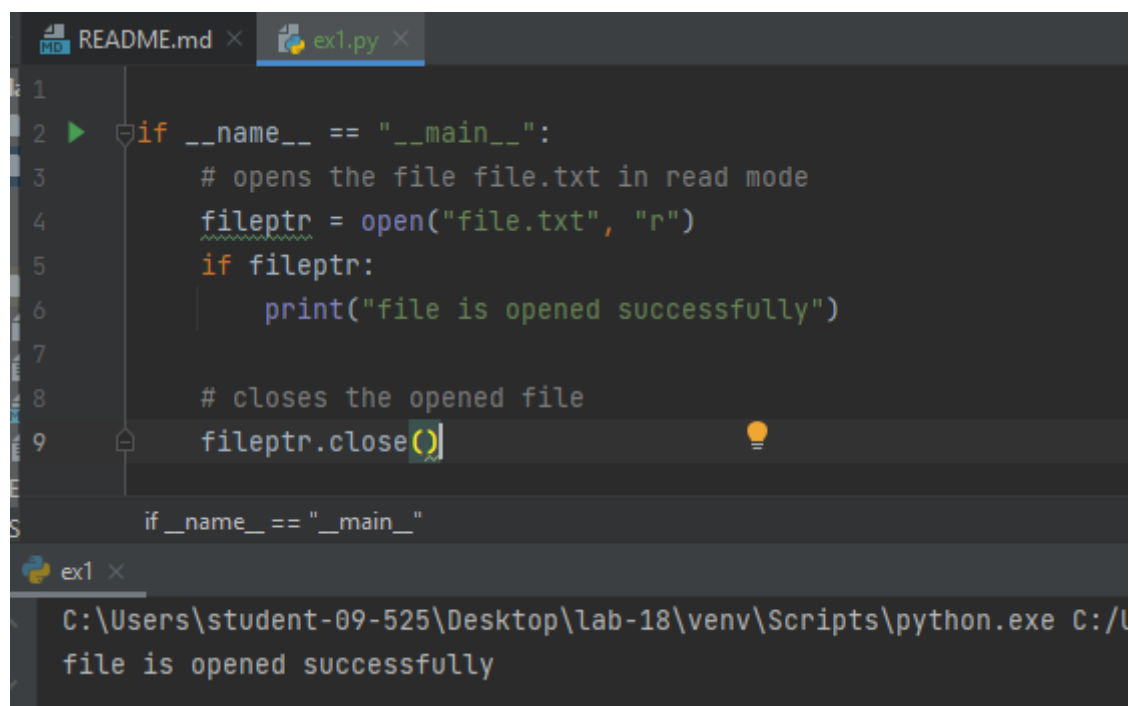


The screenshot shows a code editor with two tabs: 'README.md' and 'ex1.py'. The 'ex1.py' tab is active, displaying a Python script. The script has seven lines of code. Line 2 is a green arrow pointing to the start of the script. Line 3 is a comment. Line 4 is the opening of a file. Line 5 is the start of an if statement. Line 6 is the print statement. Line 7 is the end of the script. Below the code editor, there is a terminal window with the command 'C:\Users\student-09-525\Desktop\lab-18\venv\Scripts\python.exe C:/...' and the output 'file is opened successfully'.

```
1  
2 ► if __name__ == "__main__":  
3     # opens the file file.txt in read mode  
4     fileptr = open("file.txt", "r")  
5     if fileptr:  
6         print("file is opened successfully")  
7
```

ex1 ×
C:\Users\student-09-525\Desktop\lab-18\venv\Scripts\python.exe C:/...
file is opened successfully

Рисунок 3 – Проработка примеров

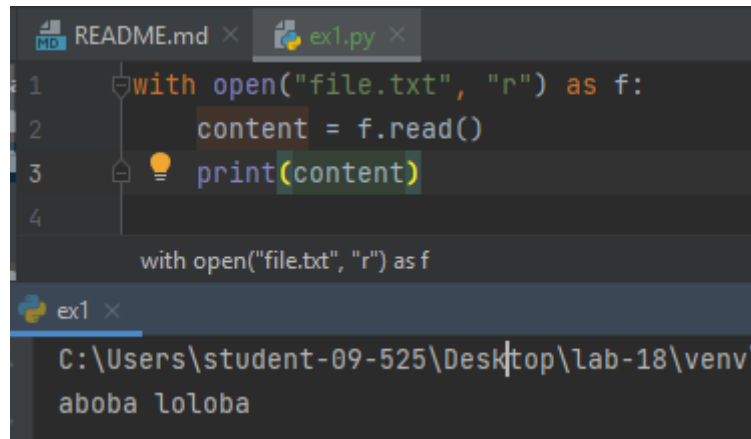


The screenshot shows a code editor with two tabs: 'README.md' and 'ex1.py'. The 'ex1.py' tab is active, displaying a Python script. The script has nine lines of code. Line 2 is a green arrow pointing to the start of the script. Line 3 is a comment. Line 4 is the opening of a file. Line 5 is the start of an if statement. Line 6 is the print statement. Line 7 is the end of the script. Line 8 is a comment. Line 9 is the closing of the file. Below the code editor, there is a terminal window with the command 'C:\Users\student-09-525\Desktop\lab-18\venv\Scripts\python.exe C:/...' and the output 'file is opened successfully'.

```
1  
2 ► if __name__ == "__main__":  
3     # opens the file file.txt in read mode  
4     fileptr = open("file.txt", "r")  
5     if fileptr:  
6         print("file is opened successfully")  
7  
8     # closes the opened file  
9     fileptr.close()
```

ex1 ×
C:\Users\student-09-525\Desktop\lab-18\venv\Scripts\python.exe C:/...
file is opened successfully

Рисунок 4 – Проработка примеров



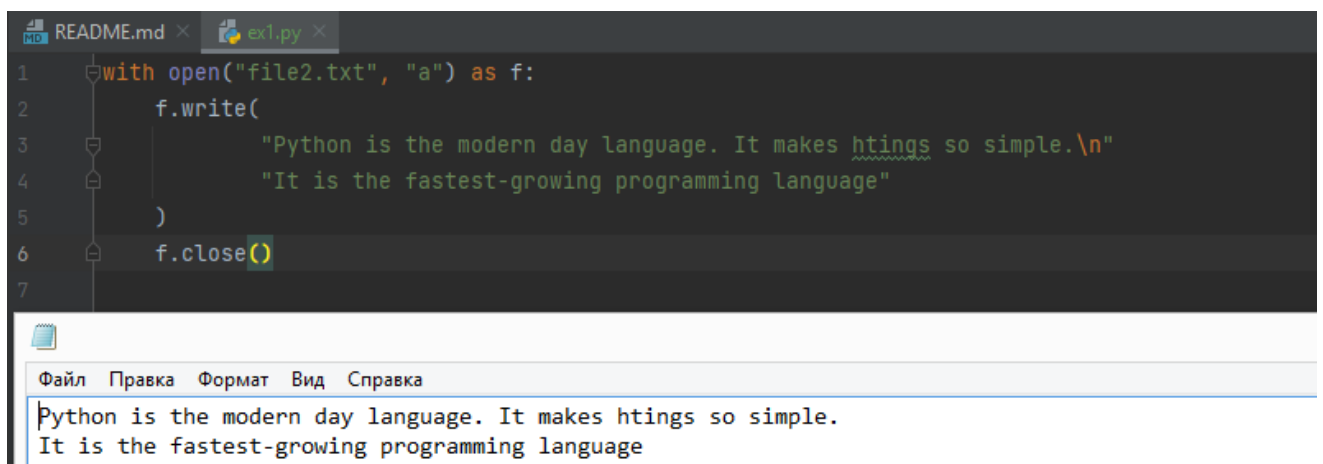
```
1 with open("file.txt", "r") as f:
2     content = f.read()
3     print(content)
4
```

with open("file.txt", "r") as f

ex1

C:\Users\student-09-525\Desktop\lab-18\venv
aboba loloba

Рисунок 5 – Проработка примеров

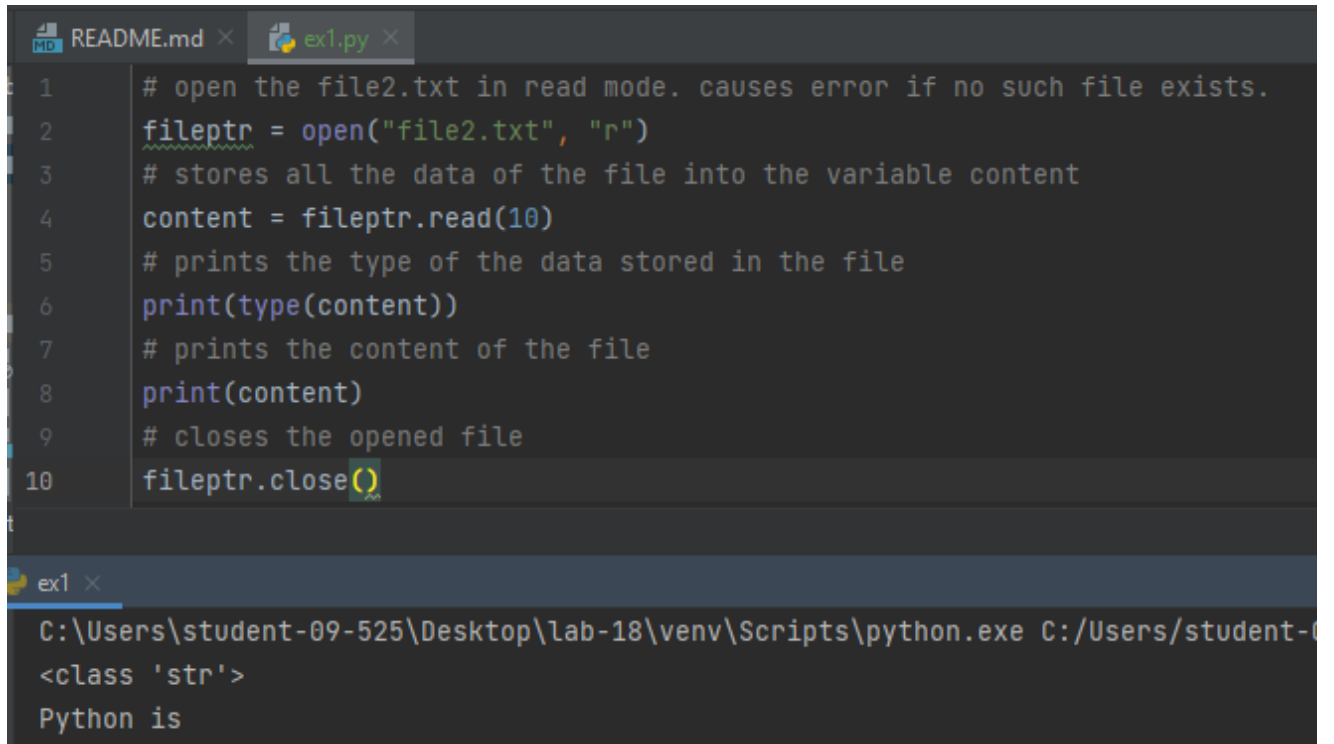


```
1 with open("file2.txt", "a") as f:
2     f.write(
3         "Python is the modern day language. It makes htings so simple.\n"
4         "It is the fastest-growing programming language"
5     )
6 f.close()
7
```

Файл Правка Формат Вид Справка

Python is the modern day language. It makes htings so simple.
It is the fastest-growing programming language

Рисунок 6 – Проработка примеров



The image shows a code editor with two tabs: 'README.md' and 'ex1.py'. The 'ex1.py' tab is active, displaying a Python script with 10 lines of code. The code opens a file named 'file2.txt' in read mode, reads 10 characters, prints the type of the data (which is a string), prints the content, and then closes the file. Below the code editor is a terminal window titled 'ex1'. It shows the command to run the script using the Python interpreter from a virtual environment. The output of the script is displayed in the terminal: '<class 'str'>' and 'Python is'.

```
1 # open the file2.txt in read mode. causes error if no such file exists.  
2 fileptr = open("file2.txt", "r")  
3 # stores all the data of the file into the variable content  
4 content = fileptr.read(10)  
5 # prints the type of the data stored in the file  
6 print(type(content))  
7 # prints the content of the file  
8 print(content)  
9 # closes the opened file  
10 fileptr.close()
```

ex1 ×
C:\Users\student-09-525\Desktop\lab-18\venv\Scripts\python.exe C:/Users/student-0
<class 'str'>
Python is

Рисунок 7 – Проработка примеров

```
1 # open the file2.txt in read mode. causes error if no such j
2 with open("file2.txt", "r") as fileptr:
3     # running a for loop
4     for i in fileptr:
5         print(i) # i contains each line of the file
6
7
```

6_ex x

C:\Users\sotni\AppData\Local\Programs\Python\Python39\python.exe C

Python is the modern day language. It makes htngs so simple.

It is the fastest-growing programming language

Рисунок 8 – Проработка примеров

```
1 # open the file2.txt in read mode. causes error if no such file exists.
2 with open("file2.txt", "r") as fileptr:
3     # stores all the data of the file into the variable content
4     content1 = fileptr.readline()
5     content2 = fileptr.readline()
6
7 # prints the content of the file
8 print(content1)
9 print(content2)
10
```

7_ex x

C:\Users\sotni\AppData\Local\Programs\Python\Python39\python.exe C:\Users\sotni\Desktop

Python is the modern day language. It makes htngs so simple.

It is the fastest-growing programming language

Рисунок 9 – Проработка примеров

```
1 # open the file2.txt in read mode. causes error if no such file exists.
2 with open("file2.txt", "r") as fileptr:
3     # stores all the data of the file into the variable content
4     content = fileptr.readlines()
5     # prints the content of the file
6     print(content)
7
```

8_ex x

C:\Users\sotni\AppData\Local\Programs\Python\Python39\python.exe C:\Users\sotni\De
['Python is the modern day language. It makes htngs so simple.\n', 'It is the fas

Рисунок 10 – Проработка примеров

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 if __name__ == "__main__":
4     # open the text.txt in append mode. Create a new file if no such file exists.
5     with open("text.txt", "w", encoding="utf-8") as fileptr:
6         # appending the content to the file
7         print(
8             "UTF-8 is a variable-width character encoding used for electronic communication.",
9             file=fileptr
10        )
11        print(
12            "UTF-8 is capable of encoding all 1,112,064 valid character code points.",
13            file=fileptr
14        )
15        print(
16            "In Unicode using one to four one-byte (8-bit) code units.",
17            file=fileptr
18        )
```

text.txt x +

Файл Изменить Просмотр

UTF-8 is a variable-width character encoding used for electronic communication.
UTF-8 is capable of encoding all 1,112,064 valid character code points.
In Unicode using one to four one-byte (8-bit) code units.

Рисунок 11 – Проработка примеров


```
1 # open the file file2.txt in read mode
2 with open("file2.txt", "r") as fileptr:
3     # initially the filepointer is at 0
4     print("The filepointer is at byte :", fileptr.tell())
5     # changing the file pointer location to 10.
6     fileptr.seek(10);
7     # tell() returns the location of the fileptr.
8     print("After reading, the filepointer is at:", fileptr.tell())
```

10_ex x

C:\Users\sotni\AppData\Local\Programs\Python\Python39\python.exe C:\Users\

The filepointer is at byte : 0

After reading, the filepointer is at: 10

Рисунок 12 – Проработка примеров

```

1      import os
2
3      # rename file2.txt to file3.txt
4      os.rename("file2.txt", "file3.txt")

```

Рисунок 13 – Проработка примеров

```

1      import os
2
3      # deleting the file named file3.txt
4      os.remove("file3.txt")
5

```

12 x

C:\Users\sotni\AppData\Local\Programs\Python\Python39\

Process finished with exit code 0

Рисунок 14 – Проработка примеров

```

1      import os
2
3      #creating a new directory with the name new
4      os.mkdir("new")

```

13 x

C:\Users\sotni\AppData\Local\Programs\Python\Python39\pyt

Рисунок 15 – Проработка примеров

```
1 import os
2
3 path = os.getcwd()
4 print(path)
```

14 x

C:\Users\sotni\AppData\Local\Programs\Python\Pyt
C:\Users\sotni\Desktop\lab-18\ex

Рисунок 16 – Проработка примеров

```
1 import os
2
3 # Changing current directory with the new directiory
4 os.chdir("C:\\Windows")
5 #It will display the current working directory
6 print(os.getcwd())
```

15 x

C:\Users\sotni\AppData\Local\Programs\Python\Python39\python
C:\Windows

Рисунок 17 – Проработка примеров

```
1 import os
2
3 # removing the new directory
4 os.rmdir("new")
```

16 x

C:\Users\sotni\AppData\Local\Programs\Python\Python39\python.exe C:\Users\sotni\Desktop\lab-18\ex\ex16.py

Process finished with exit code 0

Рисунок 18 – Проработка примеров

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import sys
5
6
7 if __name__ == "__main__":
8     print("Number of arguments:", len(sys.argv), "arguments")
9     print("Argument List:", str(sys.argv))
10
```

ex17 x

C:\Users\sotni\AppData\Local\Programs\Python\Python39\python.exe C:\Users\sotni\Desktop\lab-18\ex\ex17.py

Number of arguments: 1 arguments

Argument List: ['C:\\Users\\sotni\\Desktop\\lab-18\\ex\\ex17.py']

Process finished with exit code 0

Рисунок 19 – Проработка примеров

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import sys
5
6 ▶ if __name__ == "__main__":
7     for idx, arg in enumerate(sys.argv):
8         print(f"Argument #{idx} is {arg}")
9         print("No. of arguments passed is ", len(sys.argv))
10
```

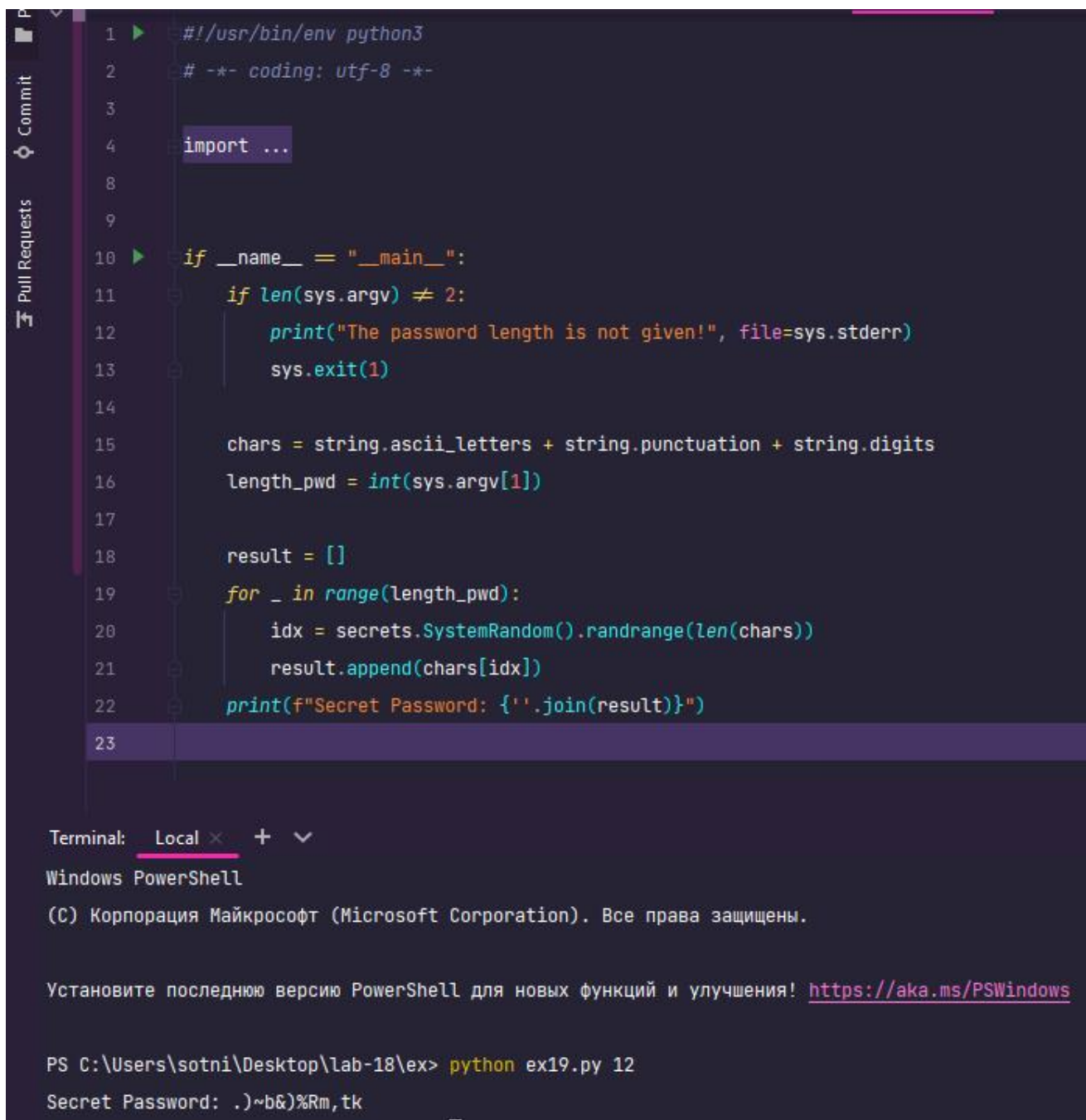
ex18 x

C:\Users\sotni\AppData\Local\Programs\Python\Python39\python.exe C:\Users\sotni\Desktop\lab-18\ex\ex18.py

Argument #0 is C:\Users\sotni\Desktop\lab-18\ex\ex18.py

No. of arguments passed is 1

Рисунок 20 – Проработка примеров



The image shows a code editor with a Python script and a terminal window below it. The script is a simple password generator. The terminal shows the command to run the script with an argument '12', and the output is a secret password.

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import ...
5
6
7
8
9
10 if __name__ == "__main__":
11     if len(sys.argv) != 2:
12         print("The password length is not given!", file=sys.stderr)
13         sys.exit(1)
14
15     chars = string.ascii_letters + string.punctuation + string.digits
16     length_pwd = int(sys.argv[1])
17
18     result = []
19     for _ in range(length_pwd):
20         idx = secrets.SystemRandom().randrange(len(chars))
21         result.append(chars[idx])
22     print(f"Secret Password: {''.join(result)}")
23
```

Terminal: Local x + v

Windows PowerShell

(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

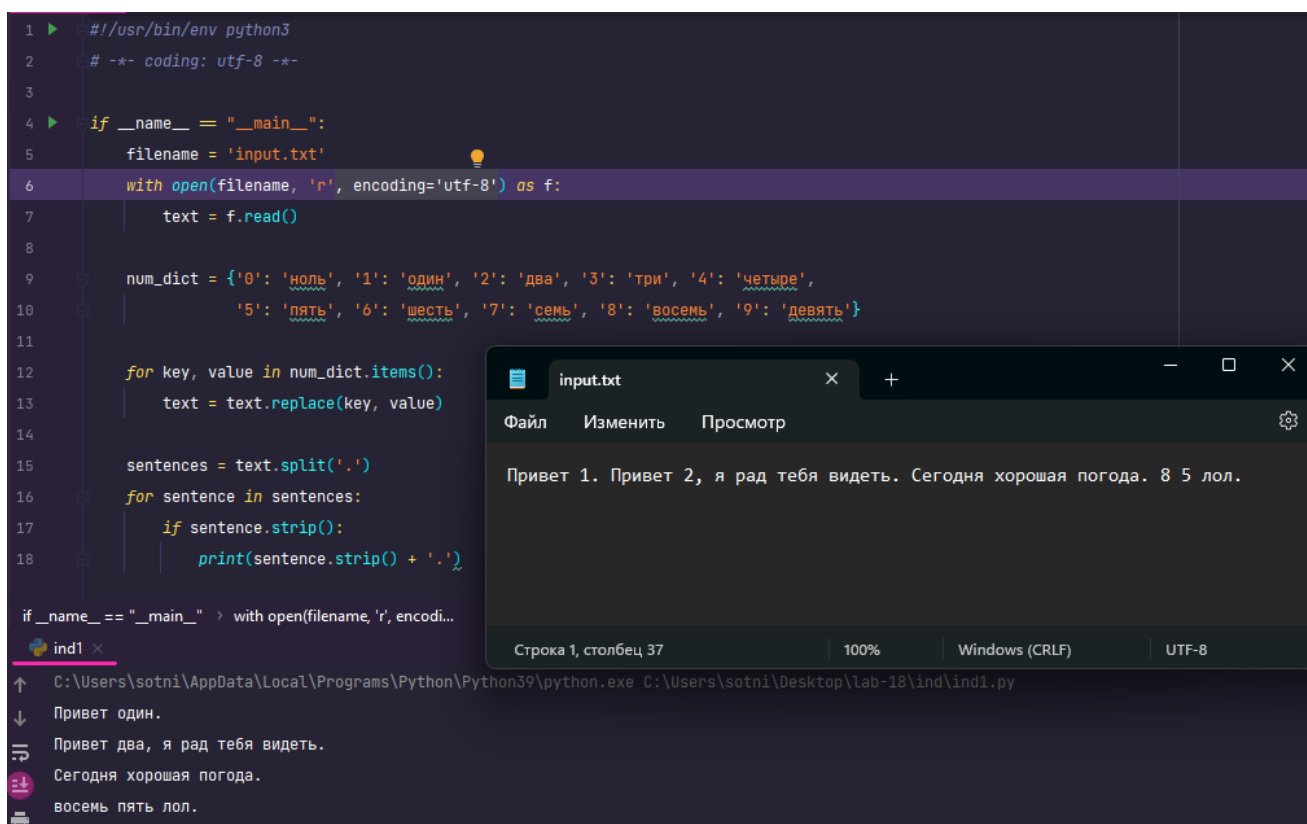
Установите последнюю версию PowerShell для новых функций и улучшения! <https://aka.ms/PSWindows>

PS C:\Users\sotni\Desktop\lab-18\ex> python ex19.py 12

Secret Password: .)~b&)%Rm,tk

Рисунок 21 – Проработка примеров

Задание №1: Написать программу, которая считывает текст из файла и выводит его на экран, заменив цифры от 0 до 9 на слова «ноль», «один», ..., «девять», начиная каждое предложение с новой строки.



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == "__main__":
5     filename = 'input.txt'
6     with open(filename, 'r', encoding='utf-8') as f:
7         text = f.read()
8
9     num_dict = {'0': 'ноль', '1': 'один', '2': 'два', '3': 'три', '4': 'четыре',
10                '5': 'пять', '6': 'шесть', '7': 'семь', '8': 'восемь', '9': 'девять'}
11
12     for key, value in num_dict.items():
13         text = text.replace(key, value)
14
15     sentences = text.split('.')
16     for sentence in sentences:
17         if sentence.strip():
18             print(sentence.strip() + '.')
19
20 if __name__ == "__main__":
21     with open(filename, 'r', encoding='utf-8') as f:
22         text = f.read()
```

input.txt

Файл Изменить Просмотр

Привет 1. Привет 2, я рад тебя видеть. Сегодня хорошая погода. 8 5 лол.

Строка 1, столбец 37 100% Windows (CRLF) UTF-8

C:\Users\sotni\AppData\Local\Programs\Python\Python39\python.exe C:\Users\sotni\Desktop\lab-18\ind\ind1.py

Привет один.

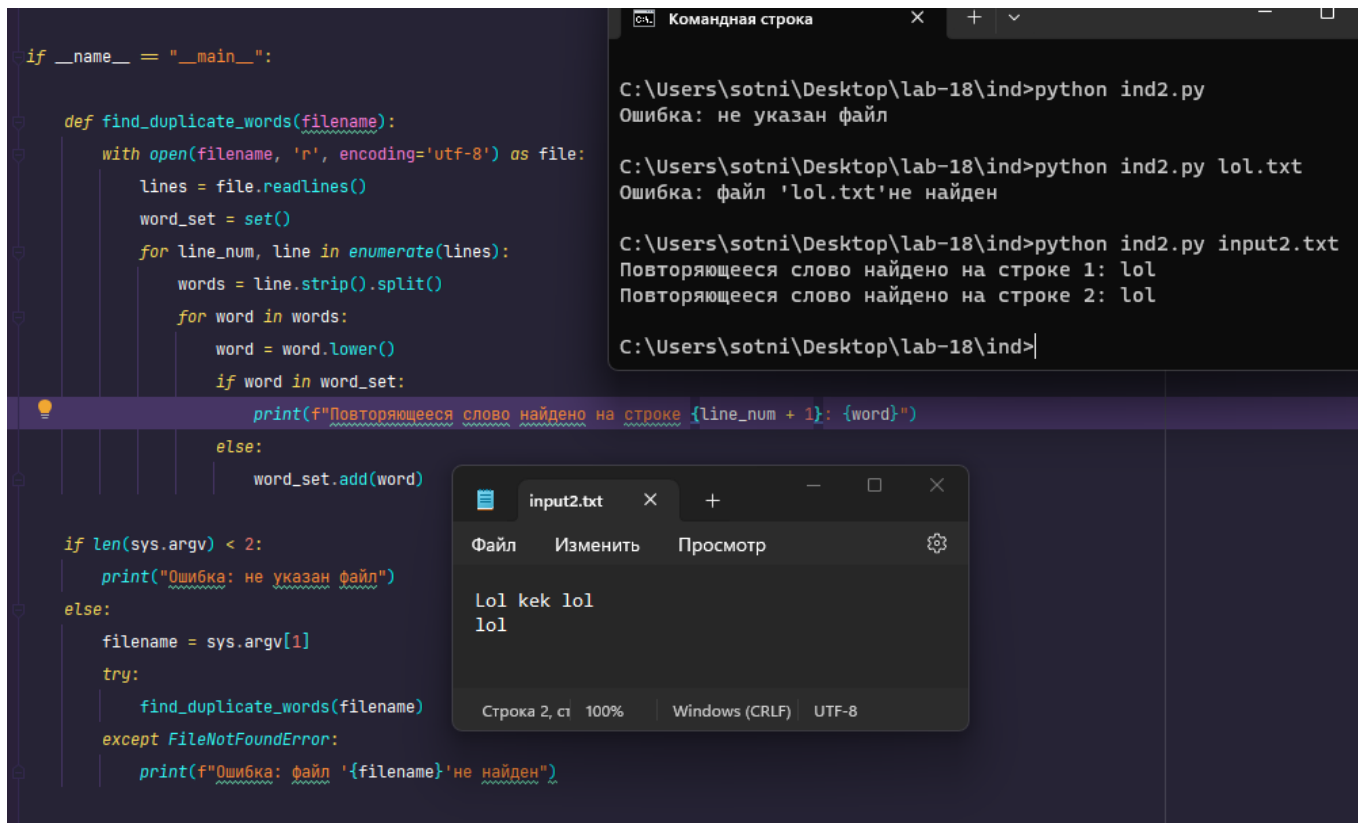
Привет два, я рад тебя видеть.

Сегодня хорошая погода.

восемь пять лол.

Рисунок 22 – Результат работы программы для первого задания

Задание №2: Проверка орфографии – лишь составная часть расширенного текстового анализа на предмет наличия ошибок. Одной из самых распространенных ошибок в текстах является повторение слов. Например, автор может по ошибке дважды подряд написать одно слово. Некоторые текстовые процессоры умеют распознавать такой вид ошибок при выполнении текстового анализа. В данном упражнении вам предстоит написать программу для определения наличия дублей слов в тексте. При нахождении повтора на экран должен выводиться номер строки и дублирующееся слово. Удостоверьтесь, что программа корректно обрабатывает случаи, когда повторяющиеся слова находятся на разных строках. Имя файла для анализа должно быть передано программе в качестве единственного аргумента командной строки. При отсутствии аргумента или невозможности открыть указанный файл на экране должно появляться соответствующее сообщение об ошибке.



```
if __name__ == "__main__":

    def find_duplicate_words(filename):
        with open(filename, 'r', encoding='utf-8') as file:
            lines = file.readlines()
            word_set = set()
            for line_num, line in enumerate(lines):
                words = line.strip().split()
                for word in words:
                    word = word.lower()
                    if word in word_set:
                        print(f"Повторяющееся слово найдено на строке {line_num + 1}: {word}")
                    else:
                        word_set.add(word)

    if len(sys.argv) < 2:
        print("Ошибка: не указан файл")
    else:
        filename = sys.argv[1]
        try:
            find_duplicate_words(filename)
        except FileNotFoundError:
            print(f"Ошибка: файл '{filename}' не найден")
```

```
C:\Users\sotni\Desktop\lab-18\ind>python ind2.py
Ошибка: не указан файл

C:\Users\sotni\Desktop\lab-18\ind>python ind2.py lol.txt
Ошибка: файл 'lol.txt' не найден

C:\Users\sotni\Desktop\lab-18\ind>python ind2.py input2.txt
Повторяющееся слово найдено на строке 1: lol
Повторяющееся слово найдено на строке 2: lol

C:\Users\sotni\Desktop\lab-18\ind>
```

input2.txt

Файл | Изменить | Просмотр

Lol kek lol
lol

Строка 2, ст 100% | Windows (CRLF) | UTF-8

Рисунок 23 – Код программы для задания №2

Задание №3: Самостоятельно подберите или придумайте задачу для работы с изученными функциями модуля os. Приведите решение этой задачи

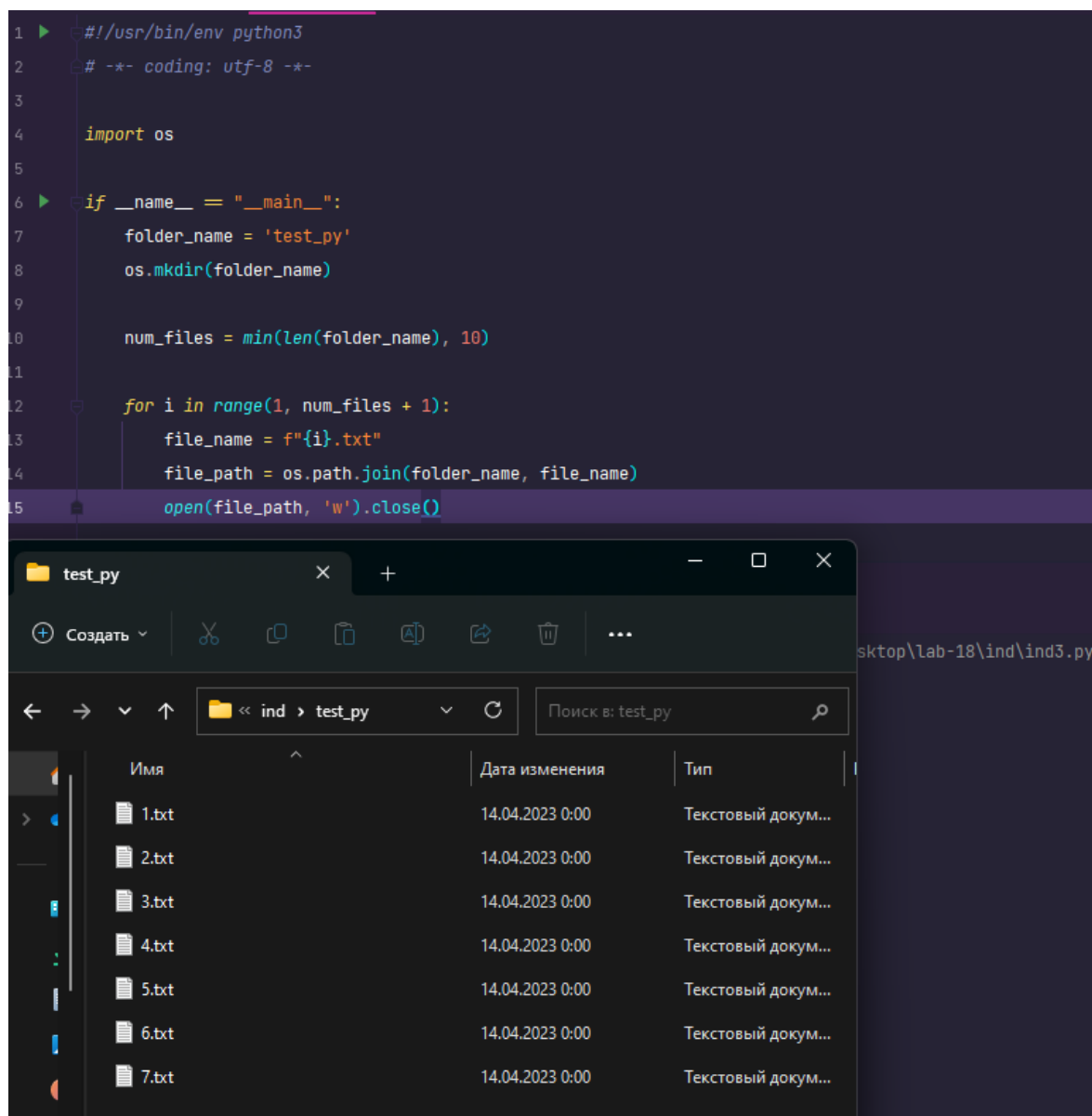


Рисунок 24 – Код и результат работы программы

Контрольные вопросы

1. Как открыть файл в языке Python только для чтения?

```
file object = open(<file-name>, <access-mode>, <buffering>)
```

Доступ к файлам можно получить с помощью различных режимов, таких как чтение, запись или добавление. Ниже приведены подробные сведения о режимах доступа для открытия файла.

- `r` – открывает файл в режиме **только для** чтения. Указатель файла существует в начале. Файл по умолчанию открывается в этом режиме, если не передан режим доступа.
- `rb` – открывает файл в двоичном формате **только для** чтения. Указатель файла существует в начале файла.

2. Как открыть файл в языке Python только для записи?

- `w` – только для записи. Он перезаписывает файл, если он существовал ранее, или создает новый, если файл с таким именем не существует. Указатель имеется в начале файла.

-
- `wb` – открывает файл для записи только в двоичном формате. Перезаписывает файл, если он существует ранее, или создает новый, если файл не существует. Указатель файла существует в начале файла.

3. Как прочитать данные из файла в языке Python?

Чтобы прочитать файл с помощью сценария Python, Python предоставляет метод `read()`. Метод `read()` считывает строку из файла. Он может читать данные как в текстовом, так и в двоичном формате.

Синтаксис метода `read()` приведен ниже.

```
fileobj.read(<count>)
```

4. Как записать данные в файл в языке Python?

Запись файла

Чтобы записать текст в файл, нам нужно открыть файл с помощью метода `open` с одним из следующих режимов доступа.

- 'w': он перезапишет файл, если какой-либо файл существует. Указатель файла находится в начале файла.
- 'a': добавит существующий файл. Указатель файла находится в конце файла. Он создает новый файл, если файл не существует.

Пример 1.

```
# open the file2.txt in append mode. Create a new file if no such file exists.
fileptr = open("file2.txt", "w")

# appending the content to the file
fileptr.write(
    "Python is the modern day language. It makes things so simple.\n"
    "It is the fastest-growing programming language"
)

# closing the opened the file
fileptr.close()
```

5. Как закрыть файл в языке Python?

Метод `close()`

После того, как все операции будут выполнены с файлом, мы должны закрыть его с помощью нашего скрипта Python, используя метод `close()`. Любая незаписанная информация уничтожается после вызова метода `close()` для файлового объекта.

Мы можем выполнить любую операцию с файлом извне, используя файловую систему, которая в данный момент открыта в Python; поэтому рекомендуется закрыть файл после выполнения всех операций.

Синтаксис использования метода `close()` приведен ниже.

```
fileobject.close()
```

6. Изучите самостоятельно работу конструкции `with ... as`. Каково ее назначение в языке Python? Где она может быть использована еще, помимо работы с файлами?

Конструкция `with ... as` используется для оборачивания выполнения блока инструкций менеджером контекста. Иногда это более удобная конструкция, чем `try...except...finally`.

Синтаксис конструкции `with ... as`:

```
"with" expression ["as" target] ("," expression ["as" target])* ":"  
suite
```

Теперь по порядку о том, что происходит при выполнении данного блока:

1. Выполняется выражение в конструкции `with ... as`.
2. Загружается специальный метод `__exit__` для дальнейшего использования.
3. Выполняется метод `__enter__`. Если конструкция `with` включает в себя слово `as`, то возвращаемое методом `__enter__` значение записывается в переменную.
4. Выполняется `suite`.
5. Вызывается метод `__exit__`, причём неважно, выполнилось ли `suite` или произошло исключение. В этот метод передаются параметры исключения, если оно произошло, или во всех аргументах значение `None`, если исключения не было.

7. Изучите самостоятельно документацию Python по работе с файлами.

Какие помимо рассмотренных существуют методы записи/чтения информации из файла?

[7. Input and Output — Python 3.11.2 documentation](#)

8. Какие существуют, помимо рассмотренных, функции модуля `os` для работы с файловой системой

[os — Miscellaneous operating system interfaces — Python 3.11.2 documentation](#)